File Permissions

- UNIX is a multi-user system. Every file and directory in your account can be protected from or made accessible to other users by changing its access permissions. Every user has responsibility for controlling access to their files.

- Permissions for a file or directory may be restricted to by types

- There are 3 type of permissions
    - r - read
    - w - write
    - x - execute = running a program

- Each permission (rwx) can be controlled at three levels:
    - u - user = yourself
    - g - group = can be people in the same project
    - o - other = everyone on the system

- File or Directory permission can be displayed by running ls –l command
    - -rwxrwxrwx

- Command to change permission
    - chmod

<u>Example:</u>

Type	User	Group	Everyone else
	rwx	rwx	rwx

-	= First dash or bit identifies the file type
---	= 2nd 3 bits defines the permission for user (file or dir owner)
---	= 3rd 3 bits defines the permission for group
---	= 4th 3 bits defines the permission for everyone else

Permissions can also be change through numerical method. Each of the permission types is represented by either a numeric equivalent:
read=4, write=2, execute=1
or a single letter:
read=r, write=w, execute=x

A permission of 4 or r would specify read permissions. If the permissions desired are read and write, the 4 (representing read) and the 2 (representing write) are added together to make a permission of 6. Therefore, a permission setting of 6 would allow read and write permissions.

Common Options
-f force (no error message is generated if the change is unsuccessful)
-R recursively descend through the directory structure and change the modes

Examples
If the permission desired for file1 is user: read, write, execute, group: read, execute, other: read, execute, the command to use would be

chmod 755 file1 or chmod u=rwx,go=rx file1

Reminder: When giving permissions to group and other to use a file, it is necessary to allow at least execute permission to the directories for the path in which the file is located. The easiest way to do this is to be in the directory for which permissions need to be granted:

chmod 711 . or chmod u=rw,+x . or chmod u=rwx,go=x .

where the dot (.) indicates this directory.

File Ownership

chown - change ownership
Ownership of a file can be changed with the chown command. On most versions of Unix this can only be done by the super-user, i.e. a normal user can’t give away ownership of their files. chown is used as below, where # represents the shell prompt for the super-user:

Syntax
chown [options] user[:group] file (SVR4)
chown [options] user[.group] file (BSD)

Common Options

-R recursively descend through the directory structure
-f force, and don’t report any errors

Examples
chown new_owner file

chgrp - change group
Anyone can change the group of files they own, to another group they belong to, with the chgrp command.

Syntax
chgrp [options] group file

Common Options
-R recursively descend through the directory structure
-f force, and don’t report any errors

Examples
% chgrp